

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

RAČUNARSTVO

**NINO ŠUPLJIKA**

**WEB MVC APLIKACIJA**

**„PRAĆENJE RADA VODNIH PUMPNIH STANICA“**

ZAVRŠNI RAD

ČAKOVEC, 2017.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

RAČUNARSTVO

**NINO ŠUPLJIKA**

**WEB MVC APLIKACIJA**

**„PRAĆENJE RADA VODNIH PUMPNIH STANICA“**

**WEB MVC APPLICATION**

**„PUMP STATION MONITORING“**

ZAVRŠNI RAD

MENTOR:

dr.sc. BRUNO TRSTENJAK

ČAKOVEC, 2017.

## SAŽETAK

*U ovom radu opisana je izrada web-aplikacije za praćenje rada vodnih pumpnih stranica koristeći MVC obrazac softverske arhitekture, koji je baziran na odvajanju pojedinih dijelova aplikacije u komponente, ovisno o njihovoj namjeni. Kao baza za izradu web-aplikacije korišten je PHP (engl. Hypertext Preprocessor) framework „Laravel“ uz potrebne popratne tehnologije kao što su HTML (engl. HyperText Markup Language) koji se koristi za strukturiranje podataka prikazanih korisniku, CSS (engl. Cascading Style Sheets) i „Bootstrap“ framework za opisivanje prikaza tih istih podataka te JavaScript s popularnim bibliotekama kao što su jQuery i slični za dodavanje raznih funkcionalnosti u aplikaciju. Za spremanje podataka korištena je MySQL baza podataka, dok su podatci u istoj kriptirani prema potrebi.*

*Sustav prijave korisnika u web-aplikaciju realiziran je na način da korisnici ne mogu sami otvarati račune, nego ih mora stvoriti isključivo administrator koji im i dodjeljuje/oduzima prava.*

*Osnovna je podjela aplikacije na „Vodoopskrbu“ i „Odvodnju“, a svaki od tih dijelova ima svoje objekte (pumpne stanice) koje unosi administrator. Svaki objekt ima svoje parametre koji su zamišljeni tako da ih administrator unosi u aplikaciju te dinamički dodjeljuje objektima, čime se omogućava da prilikom nastanka potrebe za novim mjerenjima nema potrebe za izmjenama u programu od strane programera, već se izmjena vrši od strane administratora koji ne mora biti stručan u programiranju. U aplikaciji za svaki objekt, osim mjerenja, možemo unositi i informacije o servisima, informacije o pregledu ugrađene opreme, kao i informacije o radovima na objektu. Budući da svi objekti imaju svoje troškove za komunikacije i električnu energiju, omogućeno je praćenje stavaka tako da se odabire objekt ili više objekata te se na taj način filtriraju podatci za troškove istih, a kod prikaza tih podataka imamo i automatsko izračunavanje ukupnog zbroja istih, također, oba osnovna dijela aplikacije imaju modul za korisnike koji omogućava unos planova održavanja. Sve bitne podatke korisnici mogu ispisati u PDF formatu.*

*Gotova aplikacija postavljena je na server na adresi: <http://ps-manager.ml>*

**Ključni pojmovi:** PHP, Laravel, HTML, CSS, Bootstrap, JavaScript, jQuery, MySQL

## SUMMARY

*This paper describes the creation of a web application used for monitoring the operation of pump stations using the MVC software architecture pattern, which is based on separation of individual application parts according to their purpose. The PHP (Hypertext Preprocessor) framework „Laravel“ was used as the application base along with necessary complementary technologies like HTML (HyperText Markup Language) which is used for describing the structure of data shown to the user, CSS (Cascading Style Sheets) and the „Bootstrap“ framework to stylize the data and JavaScript with popular libraries such as jQuery and others to add additional functionalities. For data storage, a MySQL database was used, while the data itself is encrypted when needed.*

*The user login system in the web application is implemented in such a way that users can not open accounts by themselves, instead only an administrator can create accounts and add or remove rights.*

*The application is divided into „Water supply“ and „Drainage“, and each of these parts has its own facilities(pump stations) that the administrator creates. Each facility has its own parameters that are created by the administrator and dynamically assigned to facilities, which allows the administrator to add new measurements to a facility without needing a programmer to change the source code. Besides measurements the application has the option to store information about services, examinations of installed equipment and information on works on the facility. All facilities have communication costs and electrical energy costs so a feature has been added that allows the tracking of these items in such a way that the user can select one or multiple facilities and filter the displayed data. While displaying this data, a summary is shown as well, which shows the total sum of the costs. Both parts of the application have a user module which allows the creation of maintenance plans. Application users can print all relevant data in PDF format.*

*The finished application is set up on a server at the address: <http://ps-manager.ml>*

**Key terms:** PHP, Laravel, HTML, CSS, Bootstrap, JavaScript, jQuery, MySQL

# SADRŽAJ

## SAŽETAK

1. UVOD	7
2. CILJ RADA I DOPRINOS.....	8
3. VODNE PUMPNE STANICE U VODOOPSKRBI.....	9
3.1. Crpke .....	10
3.2. Crpni spremnik .....	11
3.3. Strojarnica.....	11
3.4. Komandna prostorija .....	11
4. VODNE PUMPNE STANICE U ODVODNJI .....	12
5. KORIŠTENE TEHNOLOGIJE .....	13
5.1. HTML.....	13
5.2. PHP .....	14
5.3. CSS .....	14
5.4. JavaScript.....	15
5.5. MySQL .....	16
6. RAZVOJNI ALATI.....	17
6.1. Visual Studio Code.....	17
6.2. Composer.....	18
6.3. NPM.....	19
6.4. Gulp .....	19
7. PHP FRAMEWORK „LARAVEL“ .....	20
7.1. Laracasts .....	21
7.2. Preduvjeti.....	21
7.3. Instalacija i konfiguracija .....	21

7.4. Struktura direktorija i datoteka .....	22
7.5. Blade predlošci .....	23
7.6. Artisan naredbe komandne linije .....	23
7.7. Migracije.....	23
8. FRONT END FRAMEWORK „BOOTSTRAP“ .....	24
9. PROGRAMSKA IMPLEMENTACIJA.....	25
9.1. Instalacija dodatnih biblioteka i komponenti.....	25
9.2. Sustav autorizacije .....	25
9.3. Kreiranje migracija .....	26
9.4. Kreiranje modela .....	27
9.5. Kreiranje pogleda .....	28
9.6. Kreiranje kontrolera.....	29
10. APLIKACIJA „PS-MANAGER“ .....	30
11. MODUL „ADMINISTRACIJA“ .....	31
11.1. Parametri.....	31
11.2. Objekti .....	33
11.3. Korisnici .....	34
12. MODUL „VODOOPSKRBA“ .....	36
12.1. Objekti .....	36
12.2. Troškovi.....	38
12.3. Plan održavanja.....	40
13. MODUL „ODVODNJA“ .....	41
14. BAZA PODATAKA .....	42
14.1. Dijagram baze podataka .....	42
14.2. Tablica „users“ .....	43
14.3. Tablica „roles“ .....	43

14.4. Tablica „user_role“ .....	43
14.5. Tablica „password_resets“ .....	43
14.6. Tablica „pump_stations“ .....	43
14.7. Tablica „parameters“ .....	43
14.8. Tablica „pumpstation_parameter“ .....	43
14.9. Tablica „measurements“ .....	44
14.10. Tablica „parameter_measurements“ .....	44
14.11. Tablica „activities“ .....	44
14.12. Tablica „examinations“ .....	44
14.13. Tablica „parameter_examinations“ .....	44
14.14. Tablica „electricity_costs“ .....	44
14.15. Tablica „communications_costs“ .....	44
14.16. Tablica „maintenances“ .....	45
14.17. Tablica „migrations“ .....	45
14.18. Tablica „file_infos“ .....	45
15. SIGURNOST .....	46
15.1. Autentifikacija .....	46
15.2. Hashiranje lozinke .....	46
15.3. CSRF Token .....	46
15.4. Kontrola unosa podataka .....	47
16. ZAKLJUČAK .....	48
17. POPIS LITERATURE .....	49
POPIS SLIKA .....	50
POPIS KODOVA .....	51

## 1. UVOD

Budući da svakim danom sve više i više raste broj stanovnika, samim time rastu i industrijski kapaciteti te je potreba za vodom sve veća. S povećanjem količine potrošnje vode povećava se i količina onečišćenih voda.

Zbog važnosti vode u raznim granama čovjekova života, potrebno je osigurati dovoljne količine vode u naseljima, a time se podiže opći životni standard čovjeka i njegova okoliša. Kako bi se potrebna količina kvalitetne vode dovela do krajnjeg korisnika na što ekonomičniji način, potreban je funkcionalni vodoopskrbni sustav.

Kvalitetnim vodoopskrbnim sustavom osigurava se potrebna količina vode, dok je za zbrinjavanje otpadnih voda potreban kvalitetan i ispravan sustav odvodnje. Uloga je sustava odvodnje da onečišćenu vodu, uz što povoljnije sanitarne uvjete, odstrani iz ljudske blizine kako bi se na što tehnički ispravniji i ekonomičniji način pročistila i ispustila nakon pročišćavanja.

Budući da se voda koristi svugdje, a nema je u neograničenoj količini, bitno je imati ispravne i kvalitetne sustave za opskrbu i odvodnju vode.



## **2. CILJ RADA I DOPRINOS**

Cilj je ovoga rada izrada web-aplikacije koja će omogućiti informatizaciju sustava za upravljanje podacima u vezi s praćenjem rada pumpnih stanica na područjima vodoopskrbe i odvodnje. Gotova aplikacija mora imati mogućnost unosa svi potrebnih podataka o obavljenim mjerenjima i ispitivanjima, servisima, pregledima i radovima na pumpnim stanicama. Također, aplikacija mora imati mogućnost praćenja troškova, kao i izradu planova za budućnost.

Prilikom izrade rada cilj je proširiti postojeće znanje o web-aplikacijama i popratnim tehnologijama te dobiti nova znanja u vezi s radom pumpnih stanica, kao i cjelokupnog sustava za vodoopskrbu i odvodnju.

### 3. VODNE PUMPNE STANICE U VODOOPSKRBI

Vodoopskrbnim se sustavom nazivaju objekti i mjere povezane u funkcionalnu cjelinu kojima je osnovni cilj osiguranje dovoljne količine vode na što ekonomičniji način.

Vodoopskrbni sustav sastoji se od sljedećih objekata:

- vodozahvati – služe za zahvaćanje vode iz prirodnih izvora
- crpne (pumpne) stanice – služe za crpljenje i potiskivanje vode
- uređaji za kondicioniranje vode – osiguravaju zahtijevanu kvalitetu vode
- rezervoari – služe za spremanje vode
- glavna vodoopskrbna mreža – transportira vodu između objekata vodoopskrbnog sustava
- razdjelna mreža – distribuira vodu potrošačima.

Crpnom stanicom u vodoopskrbi naziva se građevina s pripadajućom elektrostrojarskom opremom kojom se voda crpi i podiže na tlačnu visinu potrebnu za osiguranje zahtijevane raspodjele vode potrošačima.

Potreba za crpnom stanicom javlja se ako izvor nema energijskog potencijala u odnosu na vodoopskrbno područje, ili je raspoloživi nedovoljan, pa ga treba postići (osigurati) na umjetni način.

U vodoopskrbnom sustavu crpna stanica služi za crpljenje vode iz bunara, rijeka, nisko položenih izvora te za potiskivanje vode u spremnik.

Crpna stanica sastoji se od sljedećih osnovnih dijelova:

- crpke
- crpni spremnik
- strojarnica
- komandna prostorija.

Crpne stanice često sadrže i transformatorsku stanicu kako bi se realizirao energetski priključak u skladu sa zahtjevima elektrodistribucijskog poduzeća.

Također, najčešće sadrže i opremu za zaštitu od vodnog udara koji se može javiti prilikom kontroliranog ili nekontroliranog zaustavljanja pogona jednog ili svih crpnih agregata, promjene opterećenja te brzog zatvaranja cjevovoda [1].

### 3.1. Crpke

Cjelokupna konfiguracija i konstrukcija crpne stanice ovisi o crpki.

Crpka odgovarajućeg kapaciteta i visine dizanja zajedno s pogonskim strojem (u pravilu elektromotorom) određene snage i postoljem čini crpni agregat.

Najčešći tip crpke koji se koristi u vodoopskrbi centrifugalna je crpka koja radi na principu transmisije centrifugalne sile na masu vode u energiju njenog strujanja (dizanja) kroz cjevovod. Centrifugalna crpka ima kapacitet do više stotina litara u sekundi, a visina dizanja može biti također do nekoliko stotina metara.



Slika 1. Visokotlačna bunarska potopna crpka [2]

### **3.2. Crpni spremnik**

Crpnim spremnikom naziva se prostor koji služi za sakupljanje i zadržavanje vode koja se prepumpava. Dimenzije spremnika ovise o režimu rada crpki i dotoka. Kod većih crpnih stanica postoji više međusobno odvojenih spremnika kako bi se omogućili popravci bez prekida rada crpki [1].

### **3.3. Strojarnica**

Strojarnicom se naziva prostor koji služi za smještaj crpnih agregata, kontrolnih instrumenata, krajeva usisnih cijevi i početaka potisnih cjevovoda s pripadajućim vodovodnim armaturama te drugom opremom [1].

Prostor strojarnice također mora sadržavati odgovarajuće otvore za komunikaciju, održavanje, montažu i demontažu opreme.

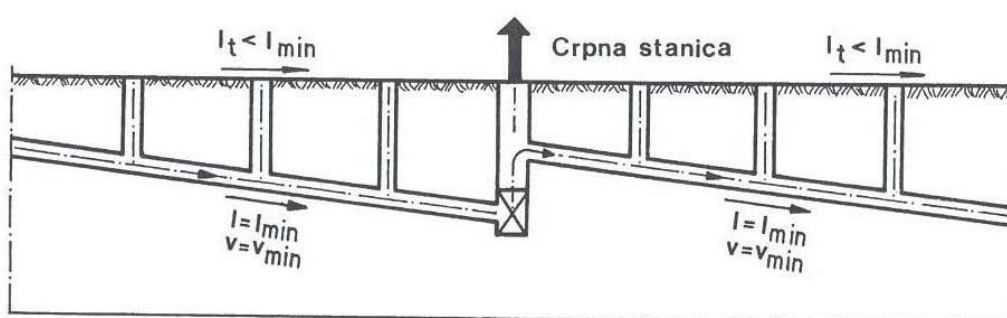
### **3.4. Komandna prostorija**

Komandnom prostorijom naziva se prostor koji sadrži potrebnu elektronsku opremu za automatsko upravljanje (ukopčavanje i iskopčavanje crpki ovisno o režimu rada) crpnim agregatima, odnosno radom crpne stanice. Također, obuhvaća i jednoliko radno opterećenje svih crpki [1].

#### 4. VODNE PUMPNE STANICE U ODVODNJI

Crpnom stanicom u odvodnji naziva se građevina s pripadajućom elektrostrojarskom opremom kojom se otpadne vode podižu iz građevina ili područja (kanala) s niže na višu razinu.

Slučajeva gdje su potrebne crpne stanice ima podosta na samoj kanalizacijskoj mreži, a ima ih i na uređajima za pročišćavanje otpadnih voda i obradu mulja. Crpne su stanice na kanalizacijskoj mreži bitne jer bi kompletna gravitacijska odvodnja zahtijevala dubine ugradnje kanala koje su daleko iznad ekonomski prihvatljivih 6 do 7 metara.



Slika 2. Shema položaja crpne stanice na kanalu [1]

Crpna stanica u odvodnji, osim osnovnih dijelova koje sadrži crpka u vodoopskrbi, mora sadržavati i rešetku pred crpnim spremnikom radi zaštite od začepljenja.

Kod odvodnje potrebna je primjena crpki koje su otporne na habanje i koroziju te crpki kojima je smanjena mogućnost začepljenja zbog toga jer voda sadrži krupnije otpatke.

Također, zbog količine otpadnih voda potrebna je crpka kapaciteta i do nekoliko kubnih metara u sekundi, dok visina dizanja mora iznositi svega nekoliko metara. Kako bi se riješio problem taloženja kod otpadnih voda, u cjevovodu se zahtijeva brzina vode između 1 i 1,5 metara u sekundi [1].

## 5. KORIŠTENE TEHNOLOGIJE

Izrađena MVC (engl. *Model-View-Controller*) web-aplikacija napravljena je na temelju MVC obrasca softverske arhitekture. Tim načinom arhitekture aplikacija se razdvaja u komponente prema njihovoj namjeni:

- model (engl. *model*) – dio aplikacije koji sadrži podatke i poslovnu logiku
- pogled (engl. *view*) – dio aplikacije koji služi za prikaz podataka iz modela
- kontroler (engl. *controller*) – dio aplikacije koji upravlja zahtjevima korisnika [3].

Te tri komponente međusobno djeluju na način da kontroler uzima potrebne podatke te ih šalje pogledu koji na temelju modela prikazuje podatke.

### 5.1. HTML

HTML (engl. *HyperText Markup Language*) osnovni je jezik za prikaz sadržaja na internetskim stranicama. Jezik je zamišljen kao besplatan i jednostavan za učenje te je zbog toga vrlo prihvaćen i popularan. Za prikaz HTML dokumenta zaslužan je web-preglednik koji čita sadržaj dokumenta te ga prikazuje korisniku. Pri izradi HTML dokumenta nastoji se težiti tome da krajnji rezultat kojeg web-preglednik prikazuje bude jednak neovisno o računalu, operativnom sustavu ili web-pregledniku. HTML nije programski jezik te kao takav ne može izvršavati nikakve operacije. Osnovi građevni element svakog HTML dokumenta su HTML oznake (engl. *tagovi*), dok svaka oznaka može imati i atribute kojima se definiraju njezina svojstva. HTML dokument je u osnovi tekstualni dokument sa „.html“ ili „.htm“ ekstenzijom.

Prva verzija HTML jezika objavljena je 1993. godine te je bila poprilično ograničena. Tijekom sljedećih godina svaki je proizvođač web-preglednika pokušao nametnuti svoje standarde za oznake, pa je dolazilo do dupliciranja oznaka (postojale su različite oznake s istim funkcijama). S verzijom HTML4 u prosincu 1997. počelo je čišćenje standarda, točnije, proglašavanje nekih od njih suvišnima. Konačna verzija HTML4.01 predstavljena je 1999. godine s manjim promjenama te je bila aktualna sve do listopada 2014. kada ju je zamijenila i trenutna inačica HTML5 s mnogim poboljšanjima [4].

## 5.2. PHP

PHP (engl. *Hypertext PreProcessor*) je open-source serverski skriptni jezik i moćan alat za izradu dinamičkih i interaktivnih web-stranica. Našao je široku primjenu kao besplatna i učinkovita alternativa konkurentskim rješenjima kao što je primjerice Microsoftov ASP.NET [5].

PHP je odlična nadogradnja na HTML tehnologiju iz razloga jer korisnik web-aplikacije ne može vidjeti kod koji je generirao sadržaj koji gleda, već mu je prikazan samo čisti HTML kod. PHP je sintaksom sličan mnogim jezicima, a jezik je bogat mnogim funkcijama za manipuliranje različitim tipova sadržaja. Ono što ga još više stavlja ispred ostalih skriptnih tehnologija, baratanje je širokom paletom baza podataka (MySQL, Oracle, PostgreSQL...) te je neovisan o operacijskom sustavu [6].

PHP skripta može se postaviti bilo gdje u dokumentu na način da počinje sa „<?php“ i završava sa „?>“, dok PHP datoteka mora imati ekstenziju „.php“.

Varijable u PHP-u uvijek počinju sa znakom „\$“ nakon čega slijedi naziv varijable, dok tip podataka nije potrebno navoditi jer PHP automatski konvertira varijablu u ispravan tip. Komentari se mogu pisati u jednom redu tako da se ispred doda „//“ ili „#“, a ukoliko se želi pisati komentar u više redova, na početku komentara potrebno je dodati „/\*“, a na kraju komentara „\*/“.

```
<?php
//Primjer ispisa teksta
$test = "Hello World";
echo $test;
?>
```

Kod 1. *Primjer ispisa teksta u PHP programskom jeziku*

*Izvor: Autor*

## 5.3. CSS

CSS (engl. *Cascading style sheets*) stilski je jezik koji se koristi za opis prezentacije HTML dokumenta. CSS-om se određuje kako će HTML elementi biti prikazani [7].

CSS pravila mogu se pisati unutar samog HTML dokumenta na način da se doda atribut „style“ sa željenim parametrima u HTML element ili unutar „<style>“ elementa u samoj

HTML stranici. Prema potrebi CSS pravila mogu se pisati i izvan HTML stranice na način da se pravila napišu unutar datoteke s ekstenzijom „.css“ te se ta datoteka pomoću HTML koda kao referenca uključi u stranicu. Referenca na vanjsku CSS datoteku mora biti unutar „<link>“ elementa koji dolazi u „<head>“ element web-stranice.

```
<head>
<link rel="stylesheet" type="text/css" href="moj_stil.css">
</head>
```

Kod 2. *Primjer referenciranja vanjske CSS datoteke u HTML jeziku*

*Izvor: Autor*

Osnovna sintaksa CSS pravila sastoji se od selektora i bloka deklaracija. Selektor označava HTML element koji se želi stilizirati, dok blok deklaracija sadrži naziv CSS svojstva te vrijednost odvojenu dvotočkom. Svaka deklaracija svojstva uvijek završava sa znakom „;“, dok blok deklaracija mora biti obavijena vitičastim zagradama. Selektori su zaslužni za pronalazak elementa za stiliziranje prema nazivu, id-u, klasi, atributu ili ostalim svojstvima. Za selektiranje tipa elementa dovoljno je samo napisati vrstu elementa, dok za selektiranje elementa po id-u moramo ispred id-a dodati znak „#“, a za selektiranje elemenata po klasi znak „.“. Moguće je i grupiranje više selektora za jedan stil, a to se radi tako da se selektori odvoje zarezom. Komentare u CSS-u moguće je pisati na način da se započne znakovima „/\*“, a završava znakovima „\*/“.

```
p, #glavni_tekst {
    color: red;
    text-align: center;
}
```

Kod 3. *Primjer CSS pravila za bojanje teksta crvenom bojom i centriranje*

*Izvor: Autor*

## 5.4. JavaScript

JavaScript je skriptni programski jezik koji se izvršava na strani korisnika, a podržavaju ga svi poznatiji preglednici. Pomoću AJAX (engl. *Asynchronous JavaScript and XML*) tehnologije omogućava nam komunikaciju između servera i korisnika, čime web-aplikacija postaje interaktivnija i lakša za korištenje [8].



Mogućnosti JavaScripta velike su jer on može čitati, pisati i mijenjati HTML elemente, može reagirati na određene događaje u pregledniku poput klika mišem, unosa teksta i sličnih. JavaScript može i provjeriti ispravnost i vjerodostojnost podataka prije slanja na server, čime poslužitelja oslobađamo dodatne obrade. Također, može i pisati i čitati kolačiće (engl. *cookies*) kao i detektirati koji preglednik koristi krajnji korisnik.

U HTML datoteku JavaScript kod može se upisati direktnim putem na način da se željeni kod piše unutar „script“ tagova. JavaScript kod također se može spremiti u zasebnu datoteku s ekstenzijom „.js“, te se taj kod može i uključiti u HTML datoteku referenciranjem putanje na tu datoteku kao „src“ atribut „script“ oznake koja se može upisati na bilo koje mjesto u HTML datoteci.

```
<head>
<script src="myScript.js"></script>
</head>
```

Kod 4. *Primjer referenciranja vanjske JavaScript datoteke u HTML jeziku*

*Izvor: Autor*

## 5.5. MySQL

MySQL besplatan je open-source sustav za upravljanje bazama podataka. MySQL baze su baze relacijskog tipa što omogućava najbolji način skladištenja podataka i brzog pretraživanja velikih količina istih što ih čini dobrim izborom za svaki informacijski sustav [9].

U MySQL bazi kao i u svakoj relacijskoj bazi svi podatci pohranjeni su unutar tablica koje se sastoje od kolona i redova, dok se odnos između tablica definira relacijama jedan prema jedan, jedan prema više, odnosno više prema jedan te više prema više.

Za upravljanje i manipuliranje bazom koristi se standardni SQL (engl. *Structured Query Language*) jezik. Neke od najvažnijih SQL naredbi su:

- SELECT – dohvaćanje podataka iz tablice
- UPDATE – mijenjanje podataka u tablici
- DELETE – brisanje podataka iz tablice
- INSERT INTO – unos podataka u tablicu
- CREATE DATABASE – kreiranje baze podataka

- ALTER DATABASE – mijenjanje baze podataka
- CREATE TABLE – kreiranje tablice
- ALTER TABLE – mijenjanje tablice
- DROP TABLE – brisanje tablice.

Naredbe pisane u SQL-u nisu „case-sensitive“, što znači da mogu biti pisane malim ili velikim slovima.

```
SELECT * FROM users;
```

Kod 5. *Primjer naredbe za odabir svih korisnika u MySQL jeziku*

*Izvor: Autor*

## 6. RAZVOJNI ALATI

Kako bi se što više olakšao i ubrzao proces izrade MVC web-aplikacije, potrebno je imati kvalitetne i moderne alate.

### 6.1. Visual Studio Code

Datoteke koje se kreiraju u tehnologijama HTML, CSS, PHP i JavaScript zapravo su sve tekstualne datoteke s pripadajućim ekstenzijama. Iz tog razloga potrebno je imati kvalitetan tekstualni editor.

Visual Studio Code je alat razvijen tek nedavno od strane tvrtke „Microsoft“. Iako je relativno nov alat, već sada ima napredne mogućnosti s kojima stvara pritisak već poznatim alatima. Glavne značajke ovog alata su:

- označavanje sintakse (engl. *syntax highlighting*) – kako bi se poboljšala čitljivost i kontekst teksta, alat različitim bojama označava različite dijelove koda (funkcije, parametre, deklaracije varijabli, zagrade)
- isječki koda (engl. *snippets*) – kako bi se ubrzalo pisanje koda koji se često ponavlja, alat omogućava pisanje koda pomoću prije definiranih isječaka koda koje možemo dodati na brz i lak način
- pametno popunjavanje koda (engl. *intellisense*) – prilikom pisanja koda alat automatski predlaže riječi koje želimo upisati na temelju toga što pišemo i njegovog prepoznavanja strukture

- debugiranje (engl. *debugging*) – pomoću raznih ekstenzija omogućeno je debugiranje aplikacije direktno iz editora
- ekstenzije – alat ima veliku podršku zajednice koja svakodnevno razvija nove korisne dodatke kojima čini alat još moćnijim
- integrirana konzola – alat ima integriranu konzolu s komandnom linijom koja omogućava lakše i brže izvršavanje naredbi potrebnih kod razvoja [10].

## 6.2. Composer

Kako se u razvoju web-aplikacija koriste i isprepliću razne tehnologije od kojih svaka ima svoje biblioteke, frameworkove i komponente, logično je da ti dijelovi imaju svoje zahtjeve, točnije komponente bez kojih ne mogu raditi.

U današnje vrijeme postoji velik broj raznih verzija raznih komponenti i teško je sve to pratiti, pa tu uskače Composer.

Composer je odličan alat kojim se upravlja zavisnostima određenog PHP projekta. Alat nema svoje korisničko sučelje, nego mu se pristupa putem konzole na način da se prije željene naredbe upiše ključna riječ „composer“.

Composer prati zavisnosti projekta putem datoteke koja ima naziv „composer.json“. Unutar te datoteke definirane su komponente koje koristi projekt te Composer sam automatski zaključuje koje su komponente zahtijevane za rad upisanih komponenti te poslaže poredak kojim se moraju učitavati.

„composer.json“ datoteka može se uređivati ručno ili putem konzole na način da se upiše ključna riječ „composer“ s naredbom „require“ i nazivom komponente.

Popis svih komponenti koje je moguće dodati putem Composera nalazi se u glavnom repozitoriju Composera nazvanom „Packagist“ koji se nalazi na internetskoj adresi <https://packagist.org/>.

```
composer require twbs/bootstrap
```

Kod 6. *Primjer naredbe za dodavanje Bootstrap front-end frameworka u projekt pomoću komandne linije*

*Izvor: Autor*

```
{
  "require": {
    "twbs/bootstrap": "3.3.7",
  }
}
```

Kod 7. *Primjer dodavanja Bootstrap front-end frameworka u projekt ručnim uređivanjem „composer.json“ datoteke*

*Izvor: Autor*

Instalacija dodanih komponenti vrši se ključnom riječi „composer“ i naredbom „install“.

### 6.3. NPM

Kako današnje web-aplikacije osim serverskog koda koriste i JavaScript jezik na klijentskom dijelu za razne funkcionalnosti, korisno je upotrebljavati i alat za upravljanje zavisnim JavaScript bibliotekama. Za tu svrhu koristi se NPM upravitelj paketa. Jednako kao i Composer, NPM koristi svoju datoteku s popisom biblioteka naziva „package.json“, a preuzimanje biblioteka vrši se naredbom „npm install“.

### 6.4. Gulp

Gulp je alat komandne linije koji služi za automatiziranje zadataka koji služe za optimizaciju rada web-aplikacija poput minifikacije JavaScript koda i CSS stilova, kao i njihovo spajanje u jednu datoteku radi bržeg učitavanja podataka zbog smanjenog broja zahtjeva prema poslužitelju.

## 7. PHP FRAMEWORK „LARAVEL“

Laravel je besplatan open-source PHP web framework kojeg je kreirao Taylor Otwell s namjerom da služi razvijanju web-aplikacija koje prate MVC (model-view-controller) obrazac softverske arhitekture. Laravel 1 objavljen je 2011. godine kao naprednija alternativa „CodeIgniter“ PHP frameworku, ali ta verzija nije imala podršku za kontrolere zbog čega se nije mogla svrstati među prave MVC frameworkove [11].

Kasnije te godine objavljen je Laravel 2 s mnogim poboljšanjima autora i zajednice te podrškom za kontrolere čime je Laravel postao punokrvni MVC framework. U toj verziji također je dodan i „Blade“ sustav za predloške s vlastitom sintaksom [11].

Početak 2012. objavljen je Laravel 3 s novim mogućnostima uključujući i „Artisan“ naredbe za komandnu liniju, bolju podršku za baze podataka te migracije za verzioniranje baze podataka. Izlaskom ove verzije porasla je popularnost i baza korisnika Laravel frameworka [11].

Laravel 4 objavljen je sredinom 2013. Ova verzija Laravela napisana je u potpunosti ispočetka te je napravljena na način da se sastoji od više zasebnih paketa koji su se distribuirali putem Composer-a. Među ostalim dodana je i podrška za inicijalno popunjavanje baze podataka (engl. *seeding*) i slanje e-mailova.

Laravel 5 je objavljen u veljači 2015., a od važnijih novosti predstavio je novu strukturu direktorija aplikacije, podršku za zadaće koje povremeno treba obaviti putem „Scheduler“ paketa te jednostavniju vanjsku autentifikaciju putem „Socialite“ paketa.

Laravel 5.1 objavljen je sredinom 2015. te je to prva verzija Laravela koja dobiva dugoročnu potporu (engl. *long term support*) s planiranim uklanjanjem nedostataka dvije godine i sigurnosnim zakrpama tri godine.

Laravel 5.3 objavljen je 23. kolovoza 2016., a nove značajke fokusirane su na ubrzavanje procesa razvoja kao i na poboljšanje općih značajki.

Sva navedena poboljšanja dovela su Laravel na sam vrh popularnosti što se tiče PHP frameworkova. Laravel također krasi prekrasno napisan kod s preciznim i jasnim komentarima te detaljnom dokumentacijom [11].

## 7.1. Laracasts

Osim detaljne dokumentacije jedan član Laravel zajednice imena Jeffrey Way pokrenuo je i web-stranicu „[www.laracasts.com](http://www.laracasts.com)“ s korisnim video isječcima vezanim uz Laravel na kojima prikazuje na koji način koristiti Laravel i povezane tehnologije te se na tim stranicama da naučiti sve potrebno za izradu Laravel web-aplikacije [12].

## 7.2. Preduvjeti

Laravel razvojni programeri pobrinuli su se da pokriju cijeli proces izrade web-aplikacija, pa su napravili i „Homestead“ virtualno računalo koje ima sve preduvjete za razvoj web-aplikacije.

No ukoliko se Laravel 5.3 aplikacija želi razvijati bez prethodno navedenog virtualnog računala, razvojno računalo mora imati predinstalirane sljedeće komponente:

- PHP verzija 5.6.4 ili novija
- OpenSSL PHP ekstenzija
- PDO PHP ekstenzija
- Mbstring PHP ekstenzija
- Tokenizer PHP ekstenzija.

Za instalaciju Laravela također je potreban i alat Composer.

## 7.3. Instalacija i konfiguracija

Kako bi se kreirala Laravel web-aplikacija, prvo je potrebno pomoću alata Composer preuzeti Laravel instalacijske datoteke.

```
composer global require "laravel/installer"
```

Kod 8. *Primjer naredbe za preuzimanje Laravel instalacije*

*Izvor: Autor*

Kada su potrebne datoteke preuzete, postupak je kreiranja aplikacije jednostavan. Pomoću komadne linije otvori se direktorij u kojem se želi kreirati aplikacija i upiše se naredba „`laravel new naziv_direktorija`“. Nakon pritiska na tipku Enter, pokrenut će se proces koji će preuzeti sve potrebne biblioteke i datoteke te će se kreirati novi direktorij upisanog naziva sa svim datotekama i poddirektorijima.

## 7.4. Struktura direktorija i datoteka

Najbitniji direktoriji i datoteke Laravel aplikacije su:

- app – sadrži klase i glavni kod aplikacije. Sastoji se od sljedećih bitnih poddirektorija:
  - Exceptions – sadrži kod za upravljanje iznimkama
  - Http – sadrži datoteke aplikacije koje služe za obradu zahtjeva poput kontroler datoteka
- config – sadrži konfiguracijske datoteke aplikacije kojima konfiguriramo baze podataka, sesije, upravljanje datotekama i slanje e-mailova
- database – sadrži datoteke vezane uz bazu podataka, a sastoji se od sljedećih bitnih poddirektorija:
  - migrations – direktorij sa datotekama koje služe za kreiranje tablica i migracija za verzioniranje baze podataka
  - seeds – direktorij sa datotekama koje služe za kreiranje inicijalnih zapisa u prethodno kreiranim tablicama baze podataka
- public – sadrži datoteke koje moraju biti dostupne krajnjem korisniku kao i „index.php“ datoteku koja služi kao ulazna točka u aplikaciju. U ovaj direktorij također se mogu dodavati i potrebne slike i fontove. Sastoji se od sljedećih bitnih poddirektorija:
  - css – sadrži CSS datoteke za stiliziranje aplikacije
  - js – sadrži JavaScript datoteke
- resources – sadrži glavne datoteke u izvornom obliku koje se tek nakon obrade prikazuju korisniku. Sastoji se od sljedećih bitnih poddirektorija:
  - assets – sadrži Javascript, CSS i ostale datoteke koje će se nakon minifikacije i zapakiravanja, te prebacivanja u „public“ direktorij prikazivati krajnjem korisniku
  - lang – sadrži datoteke koje se koriste za implementaciju višejezičnosti
  - views – sadrži izvorne „view“ datoteke
- routes – sadrži „web.php“ datoteku koja sadrži sve rute aplikacije koje će biti dostupne krajnjim korisnicima

- storage – sadrži sve datoteke koje generira Laravel framework. Sastoji se od sljedećih bitnih poddirektorija:
  - app – sadrži datoteke generirane od strane aplikacije
  - public – sadrži datoteke koje je kreirao krajnji korisnik i koje moraju biti javno dostupne
  - logs – sadrži zapisnik rada aplikacije
- tests – sadrži automatizirane testove
- vendor – sadrži zavisne PHP biblioteke kreirane od strane Composer alata.

### 7.5. Blade predlošci

Blade je jednostavan, ali moćan mehanizam za izradu predložaka. Iako ostali popularni mehanizmi za izradu predložaka onemogućuju korištenje čistog PHP koda, Blade dopušta korištenje istog u „view“ datotekama. Sve Blade „view“ datoteke se zapravo kasnije prevode u čisti PHP kod. Blade predlošci koriste „blade.php“ ekstenziju.

Dvije osnovne prednosti Blade predložaka su sekcije i nasljeđivanje predložaka.

### 7.6. Artisan naredbe komandne linije

Artisan je sučelje komandne linije koje pruža velik broj korisnih naredbi koje pomažu prilikom razvoja aplikacije. Sve Artisan naredbe pišu se s prefiksom „php artisan“. Popis naredbi moguće je pregledati pokretanjem Artisan naredbe „list“, dok se opis određene naredbe može vidjeti dodavanjem ključne riječi „help“ na kraj željene naredbe.

### 7.7. Migracije

Laravel ima ugrađen sustav za verzioniranje baze podataka pomoću migracija. Migracije su zamišljene na način da se prilikom izmjena u bazi podataka kreira kod pomoću kojeg se ostvaruju promjene, kao i kod kojim se te promjene poništavaju. Ukoliko se za svaku promjenu kreira takav kod, tada je moguće na jednostavan način ažurirati bazu podataka koju koristi korisnik, kao i poništiti promjene i vratiti se na stariju verziju baze podataka. Informacije o migracijama pohranjuju se u tablicu u bazi podataka te se pomoću njih uvijek zna koji korisnik ima koju verziju baze podataka.



## 8. FRONT END FRAMEWORK „BOOTSTRAP“

Bootstrap je jedan od najpopularnijih open-source HTML, CSS i JavaScript front-end frameworka za razvoj web-stranica. Podržava responzivni dizajn što omogućava da se izgled stranice dinamički mijenja ovisno o vrsti uređaja i razlučivosti zaslona. Bootstrap sadrži mnogo predložaka za razne komponente sučelja, a neki od najvažnijih pokrivaju sljedeće elemente:

- tablice
- gumbe
- navigaciju
- forme
- modalne prozore
- straničenje
- padajuće izbornike [13].

Bootstrap također sadržava i opcionalne JavaScript ekstenzije kojima se dodaju zanimljive mogućnosti, a također ima vrlo aktivnu zajednicu koja stvara mnoge dodatke, a zbog svoje popularnosti i mnoge biblioteke koda za razne komponente imaju kreirane posebne teme kojima se izgled usklađuje s Bootstrap principima.

Bootstrap podržava najnovije verzije Google Chrome, Firefox, Internet Explorer, Opera i Safari Internet preglednika. Razvijen je od strane Twittera, a prva verzija izdana je 19. kolovoza 2011. godine, dok je trenutna stabilna verzija (3.3.7) izdana 25. srpnja 2016. godine. Trenutno je u razvoju Bootstrap verzija 4 koja je skoro u potpunosti napisana ispočetka čime su mnogi postojeći problemi riješeni odmah na početku [13].

## 9. PROGRAMSKA IMPLEMENTACIJA

Nakon kreiranja osnovne Laravel aplikacije, konfiguriranja baze podataka i ostalih početnih postavki, može se krenuti u implementaciju aplikacije.

### 9.1. Instalacija dodatnih biblioteka i komponenti

Aplikacija koristiti sljedeće dodatne biblioteke i komponente:

- Bootstrap front-end framework
- Smartmenu bootstrap
- jQuery.

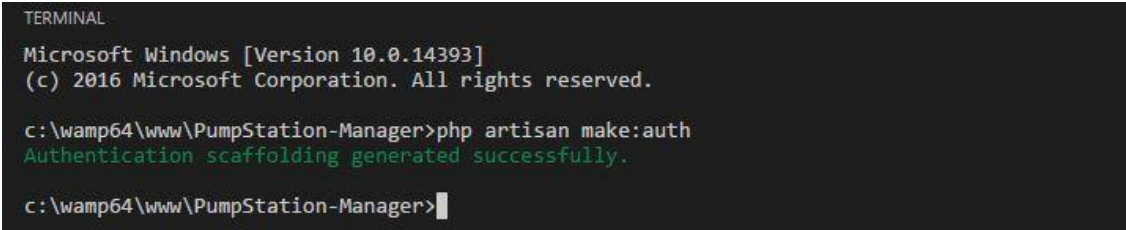
Početna konfiguracija Laravela ima podršku za navedene tehnologije te ih je samo potrebno instalirati putem NPM i Gulp alata.

Kako datoteka „package.json“ koju koristi NPM manager dolazi već konfigurirana s osnovnom instalacijom Laravela, potrebno je pokrenuti naredbu „npm install“ u komandnoj liniji kojom se preuzimaju sve potrebne datoteke.

Preuzete datoteke spajaju se u jednu cjelinu na način da se u komandnoj liniji pokrene naredba „gulp“.

### 9.2. Sustav autorizacije

Kreiranje osnovnog sustava autorizacije u Laravelu prilično je jednostavno. Potrebno je pokrenuti naredbu „php artisan make:auth“. Pokretanje te naredbe kreirat će sustav autorizacije sa svim osnovnim datotekama potrebnim za rad istog.



```
TERMINAL
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\wamp64\www\PumpStation-Manager>php artisan make:auth
Authentication scaffolding generated successfully.

c:\wamp64\www\PumpStation-Manager>
```

Slika 3. Uspješno kreiranje osnovne autentifikacije u Laravel frameworku

Izvor: Autor

Kako sustav autorizacije kreiran od strane Laravela nema podršku za korisničke uloge, potrebno ga je nadograditi. Sustav autorizacije automatiziranim postupkom kreira

migraciju kojom se kasnije u bazi podataka može kreirati tablica s korisnicima „users“ i tablica „password\_resets“ koja služi kao pomoć kod resetiranja lozinke. Da bi se dodala podrška za uloge, potrebno je napraviti tablicu s ulogama i tablicu gdje će se korisnici povezivati s tim ulogama, kao i napisati potreban kod za ostvarivanje te funkcionalnosti.

### 9.3. Kreiranje migracija

Za kreiranje svih tablica u bazi podataka korištene su migracije.

Kreiranje migracije vrši se pomoću Artisan naredbe „make:migration *naziv*“ koja kreira novu datoteku s klasom koja proširuje klasu „Migration“ koja je dio Laravel frameworka, a sadržava funkciju „up()“ u koju se piše kod za kreiranje promjena u bazi podataka i funkciju „down()“ u koju se piše kod za poništavanje promjena. Za svaku tablicu u aplikaciji kreirane su migracije prilikom kreiranja modela.

Nakon kreiranja migracija potrebno je napraviti pripadajuće promjene u bazi podataka. Pokretanjem Artisan naredbe „migrate“ izvršavaju se promjene u bazi podataka, dok se pokretanjem naredbe „migrate:rollback“ poništavaju promjene.

```
class CreateMeasurementsTable extends Migration{
    public function up() {
        Schema::create('measurements', function(Blueprint $table)
        {
            $table->increments('id');
            $table->timestamps();
            $table->integer('pumpstation_id')->unsigned();
            $table->date('measurement_date');
            $table->foreign('pumpstation_id')->references('id')
                ->on('pump_stations')->onDelete('cascade')
                ->OnUpdate('cascade');
        });
    }
    public function down() {
        Schema::drop('measurements');
    }
}
```

Kod 9. Migracija za kreiranje tablice za mjerenja parametara pumpnih stanica

Izvor: Autor

## 9.4. Kreiranje modela

Kreiranje modela vrši se pomoću Artisan naredbe „make:model“. Kako bi se prilikom modela automatski kreirala i potrebna migracija, naredbi je potrebno dodati sufiks „-m“.

Nakon kreiranja modela Laravel framework automatskim putem pomoću naziva modela i poljima kreiranim u migraciji zaključuje koja svojstva ima model. Iz toga razloga u modelima aplikacije su u većini slučajeva dodane samo veze između objekata u obliku funkcija, kao i funkcije koje obrađuju podatke za prikaz krajnjim korisnicima.

Kod 10. prikazan ispod prikazuje model klase „Activity“, a sadržava sljedeće funkcije:

- funkciju „pumpStations()“ – vraća pumpnu stanicu vezanu uz tu aktivnost
- funkciju „uploaded\_files()“ – vraća datoteke vezane uz tu aktivnost
- funkciju „boot()“ – prilikom brisanja aktivnosti briše vezane datoteke.

```
class Activity extends Model
{
    protected $fillable = ['activity_date', 'note',
                           'activity_type', 'pumpstation_id',
                           'done'];

    public function pumpStations() {
        return
            $this->belongsTo('App\PumpStation', 'pumpstation_id');
    }

    public function uploaded_files() {
        return $this->morphMany('App\FileInfo', 'uploadable');
    }

    protected static function boot() {
        parent::boot();
        static::deleting(function($item) {
            $item->uploaded_files()->delete();
        });
    }
}
```

Kod 10. Model klasa „Activity“ koja predstavlja servise, radove ili preglede ugrađene opreme na pumpnim stanicama

Izvor: Autor

## 9.5. Kreiranje pogleda

Pogledi (engl. *View*) u aplikaciji kreirani su pomoću Blade predložaka kako bi se što više smanjilo ponavljanje koda. Napravljen je jedan glavni „master“ pogled koji sadržava više pogleda, a podijeljen je na sljedeće osnovne dijelove:

- osnovne JavaScript skripte i CSS stilovi koji se uvijek učitavaju
- glavni izbornik koji se generira dinamički ovisno o podacima u bazi podataka
- dio za poruke i obavijesti koje je potrebno prikazati korisniku
- glavni sadržaj
- zaglavlje
- dodatne JavaScript skripte koje se učitavaju prema potrebi.

```
<!DOCTYPE html>
<html>

<head>
  <base href="{{ url('/') }}">
  <meta charset="utf-8">
  <title>@yield('title')</title>
  @include('layouts.master.resources')
</head>
<body>
  @include('layouts.master.main_menu')
  <div id="loadingDiv" style=""></div>
  <div id="content" class="container" style="display:none;">
    @include('layouts.master.flash_messages')
    @yield('content')
  </div>
  @include('layouts.master.footer')
  @yield('additional_scripts')
</body>

</html>
```

Kod 11. *HTML kod „master“ pogleda*

*Izvor: Autor*

## 9.6. Kreiranje kontrolera

Kontroleri u aplikaciji kreirani su pomoću Artisan naredbe „make:controller *naziv*“. Svaki kontroler proširuje klasu „Controller“ koja je dio Laravel frameworka. Kontroler se sastoji od funkcija koje vraćaju neki rezultat koji, između ostalog, može biti pogled ili datoteka.

```
class UserController extends Controller
{
    public function show($id)
    {
        return view('user.profile',
                    ['user' => User::findOrFail($id)]);
    }
}
```

Kod 12. *Primjer kontrolera za prikaz podataka određenog korisnika*

*Izvor: Autor*

Kako bi se pozvala funkcija u kontroleru, potrebno je kreirati rute koje će prilikom zahtjeva od strane korisnika voditi do funkcija. Popis postojećih ruta nalazi se u datoteci „routes/web.php“. Prilikom kreiranja ruta potrebno je uzeti u obzir tip korisnika koji može pristupiti tim rutama.

```
Route::get('/administration/users',
           'AdministrationController@users_index')
    ->name('users_index')
    ->middleware('roles:Administrator');

Route::get('/administration/users/edit/{user?}',
           'AdministrationController@users_edit')
    ->name('users_edit')
    ->middleware('roles:Administrator');

Route::post('/administration/users/edit/{user?}',
            'AdministrationController@users_edit')
    ->name('users_edit')
    ->middleware('roles:Administrator');

Route::get('/administration/users/delete/{user}',
           'AdministrationController@users_delete')
    ->name('users_delete')
    ->middleware('roles:Administrator');
```

Kod 13. *Rute za upravljanje korisnicima*

*Izvor: Autor*

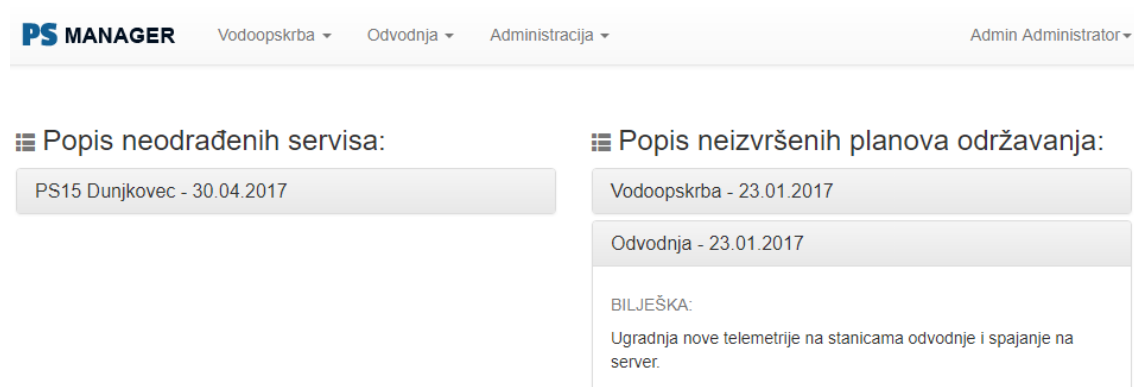
## 10. APLIKACIJA „PS-MANAGER“

Aplikacija se sastoji od 3 modula:

- **administracija** – upravljanje i pregled korisnika aplikacije, objekata i parametara
- **vodoopskrba** – upravljanje i pregled mjerenja, servisa, pregleda ugrađene opreme i radova na vodoopskrbnim objektima, troškovima i planovima održavanja
- **odvodnja** – upravljanje i pregled mjerenja, servisa, pregleda ugrađene opreme i radova na objektima odvodnje, troškovima i planovima održavanja.

U aplikaciji postoje 3 vrste korisničkih prava:

- **administrator** – može upravljati podacima svih modula
- **djelatnik** – može upravljati podacima u modulima vodoopskrbe i odvodnje
- **gost** – može samo pregledavati podatke u modulima vodoopskrbe i odvodnje.



Slika 4. Početna stranica aplikacije prijavljenog korisnika

Izvor: Autor

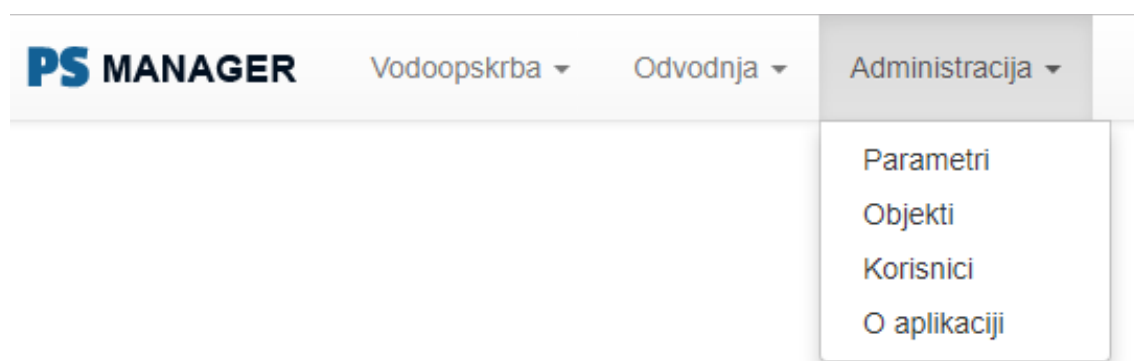
Upravljanje zapisima u bazi podataka i pretraživanjem usklađeno je u cijeloj aplikaciji te postupci opisani u sljedećem poglavlju vrijede za sve module.

## 11. MODUL „ADMINISTRACIJA“

Modul „Administracija“ omogućava upravljanje sljedećim podacima aplikacije:

- parametri
- objekti
- korisnici.

U administratorski dio aplikacije može ući samo korisnik tipa „Administrator“ klikom na gumb „Administracija“ te izborom stavke koju želi uređivati.



Slika 6. Izbornik modula „Administracija“

Izvor: Autor

### 11.1. Parametri

U aplikaciji postoji mogućnost dodavanja dvije vrste parametara:

- mjerenje – predstavlja parametar kojeg možemo mjeriti nekim instrumentom (npr. napon, jakost struje)
- ispitivanje – predstavlja parametar/komponentu koja se ispituje (npr. osigurač, snaga pumpe).

Klikom na gumb u izborniku „Parametri“ pojavljuje se popis postojećih parametara. Popis parametara moguće je filtrirati unosom pojma za pretragu te klikom na gumb „Traži“. Pokraj gumba „Traži“ nalazi se gumb za osvježavanje popisa parametara.

Ukoliko je korisnik tipa „Administrator“ ili „Djelatnik“, tada on ima mogućnost dodavanja zapisa klikom na gumb „+“ ili brisanja zapisa klikom na gumb „x“ u pripadajućem retku na popisu.



## Administracija parametara

PRETRAŽIVANJE

Traži



### Popis parametara

Naziv	Vrsta	
Id (mA)	Mjerenje	✗
Idn (A)	Mjerenje	✗
In (A)	Mjerenje	✗
In (A) termičke zaštite	Ispitivanje	✗
Instalirana snaga ili limitator (A)	Ispitivanje	✗
Izmjerena struja pumpe (A)	Ispitivanje	✗
Komunikacija (GPRS,SMS)	Ispitivanje	✗
Neprekinutost zaštitnog vodiča (ohm)	Mjerenje	✗
Osigurač pumpi	Ispitivanje	✗
Otpor petlje dolaznog vodiča (ohm)	Mjerenje	✗

« 1 2 3 »

Slika 7. Prikaz popisa parametara u aplikaciji

Izvor: Autor

Prilikom dodavanja ili uređivanja parametra obavezni podatci za unos su naziv i vrsta parametra. Klikom na gumb „Spremi“ postojeći ili novi zapis sprema se u bazu podataka. Klikom na gumb „Povratak“ vraćamo se na popis parametara.

## Dodavanje novog parametra

Naziv

Vrsta

Mjerenje

▼

Povratak

Spremi

Slika 8. Dodavanje novog parametra

Izvor: Autor

## 11.2. Objekti

Objekti predstavljaju postrojenje koje ima neku funkciju, a mogu biti vezani uz vodoopskrbu ili odvodnju.

Ukoliko se radi o objektu vodoopskrbe, tada postoji samo mogućnost dodjeljivanja parametara mjerenja, dok kod objekta odvodnje postoji mogućnost dodjeljivanja parametara mjerenja i ispitivanja.

## Administracija objekata

PRETRAŽIVANJE

Q Traži



### Popis objekata

Naziv	Vrsta	
Crpilište Nedelišće	Vodoopskrba	×
Crpilište Prelog	Vodoopskrba	×
Crpilište Sv.Marija	Vodoopskrba	×
PCP Lopatinec	Vodoopskrba	×
PS Ivanovec Prelog	Odvodnja	×
PS1 Ivanovec	Odvodnja	×
PS1 Novo Selo Rok	Odvodnja	×
PS14 Dunjkovec	Odvodnja	×
PS15 Dunjkovec	Odvodnja	×
PS2 Ivanovec	Odvodnja	×

Slika 9. Prikaz popisa objekata u modulu „administracija“

Izvor: Autor

Kod unosa ili uređivanja objekta moguće je unijeti zadane vrijednosti za komunikacijske troškove (broj telefona, napomena), kao i zadane vrijednosti za troškove električne energije (broj obračunskog mjernog mjesta).

## Uređivanje postojeće pumpne stanice

Naziv	<input type="text" value="PS2 Ivanovec"/>
Kategorija objekta	<input type="text" value="Odvodnja"/>
Mjerenja	<input type="text" value="× Id (mA) × Idn (A) × Uc (V)"/>
Ispitivanja	<input type="text" value="× Instalirana snaga ili limitator (A) × Osigurač pumpi"/>
Komunikacijski troškovi	<input checked="" type="checkbox"/>
Broj telefona	<input type="text" value="098123456"/>
Napomena (Komunikacije)	<input type="text" value="pretplata po ugovoru br. 123456 (Hrvatski telekom)"/>
Troškovi el. energije	<input checked="" type="checkbox"/>
Broj OMM	<input type="text" value="12345"/>
	<input type="button" value="Povratak"/> <input type="button" value="Spremi"/>

Slika 10. Uređivanje postojeće pumpne stanice

Izvor: Autor

### 11.3. Korisnici

Inicijalno aplikacija dolazi s jednim korisnikom tipa administrator, sa sljedećim podacima:

- e-mail: admin@ps-manager.tk
- lozinka: 123456

Taj korisnik tada može kreirati nove korisnike te im dodjeljivati prava. Prilikom prvog pokretanja aplikacije preporučljivo je promijeniti podatke administratorskog računa.

## Administracija korisnika

PRETRAŽIVANJE



### Popis korisnika

Ime i Prezime	E-mail	Gost	Djelatnik	Administrator
Ana Anić	aanic@mev.hr	—	✓	—
Ivo Ivić	iivic@mev.hr	✓	—	—
Nino Šupljika	nsupljika@student.mev.hr	—	—	✓
Pero Perić	pperic@mev.hr	—	—	✓

Slika 11. Pregled popisa korisnika aplikacije

Izvor: Autor

Prilikom kreiranja novog korisnika potrebno je unijeti osnovne podatke: ime, prezime, korisnička prava te e-mail i lozinku koju će korisnik koristiti za prijavu.

## Dodavanje novog korisnika

Ime	<input type="text" value="Pero"/>
Prezime	<input type="text" value="Perić"/>
E-mail	<input type="text" value="pperic@mev.hr"/>
Lozinka	<input type="password" value="....."/>
Ponovljena lozinka	<input type="password" value="....."/>
Korisnička prava	<input type="text" value="✖ Djelatnik"/>
<input type="button" value="Povratak"/> <input type="button" value="Spremi"/>	

Slika 12. Dodavanje novog korisnika u aplikaciju

Izvor: Autor

## 12. MODUL „VODOOPSKRBA“

Modul „Vodoopskrba“ omogućava upravljanje sljedećim podacima aplikacije:

- praćenje rada objekata vodoopskrbe
- troškovima vodoopskrbe
  - električna energija
  - komunikacije
- planovima održavanja za vodoopskrbu.

Sve zapise iz ovog modula moguće je prikazati u PDF obliku prema odabranim parametrima pretrage klikom na ikonu pisača na pripadajućem mjestu.

### 12.1. Objekti

Ukoliko se odabere objekt u izborniku modula vodoopskrbe prikazuje se izbornik objekta sa sljedećim stavkama:

- mjerenja
- servisi
- pregled ugrađene opreme
- radovi na objektu.

Klikom na „Mjerenja“ otvara se popis mjerenja izvršenih na objektu. Pretraživanje zapisa omogućeno je po datumu, dok se parametri za prikaz na popisu pojavljuju dinamički ovisno o odabiru u pretraživaču. Ispis mjerenja je također dinamičan te se prikazuju samo parametri odabrani prilikom pretrage.

Prilikom unosa novog mjerenja moguće je unijeti izmjerene vrijednosti za svaki parametar.

## Crpilište Nedelišće

Mjerenja
Servisi
Pregled ugrađene opreme
Radovi na objektu
Ispitivanja i pregled

PRETRAŽIVANJE

Mjerenja

Od  Do

Popis obavljenih mjerenja

Datum	Id (mA)	Idn (A)	In (A)	Presjek dolaznog vodiča	Ra (ohm)	t (ms)	Uc (V)	Zaštitni uređaj	
24.05.2016	225	0,3	25	5x10mm2	3,28	127	1	FID 25/0,3	✖

Slika 13. Prikaz popisa obavljenih mjerenja na pumpnoj stanici

Izvor: Autor

Klikom na „Servisi“ u izborniku pojedinog objekta prikazuje se popis servisa. Servisi se mogu označiti kao obavljeni klikom na gumb „Servis nije odrađen“. Ukoliko servis nije odrađen, on se također prikazuje na početnoj stranici prijavljenog korisnika kako bi se omogućio brži pregled stanja.

## Crpilište Nedelišće

Mjerenja
Servisi
Pregled ugrađene opreme
Radovi na objektu
Ispitivanja i pregled

PRETRAŽIVANJE

Od  Do

Evidencija servisa

16.03.2016

BILJEŠKA:

Crpka 2 poslana na servis. Radni sti 5380h.  
Popravak izvršio VDG Strahoninec.

✖

✔ Servis je odrađen.

Slika 14. Prikaz evidencije servisa pumpne stanice

Izvor: Autor

Prilikom dodavanja ili uređivanja servisa potrebno je upisati bilješku, a moguće je i učitavanje novih ili brisanje postojećih datoteka bilo kojeg formata klikom na gumb „Odaberi datoteku“ za dodavanje ili „x“ pokraj naziva datoteke za brisanje.

## Crpilište Nedelišće

### Dodavanje novog zapisa evidencije servisa

The screenshot shows a web form for adding a new service record. It includes a date field set to 11.09.2017, a text area for a note containing 'Crpka 1 poslana na servis. Radni sati 5082h. Popravak izvršava - VDG Strahoninec', and a file upload section with a button 'Odaberi datoteku' and a message 'Nije odabrana niti jedna datoteka.' At the bottom are 'Povratak' and 'Spremi' buttons.

Datum	11.09.2017
Bilješka	Crpka 1 poslana na servis. Radni sati 5082h. Popravak izvršava - VDG Strahoninec
Datoteke	<div> Odaberi datoteku Nije odabrana niti jedna datoteka.</div> <div>Povratak Spremi</div>

Slika 15. Uređivanje postojećeg zapisa evidencije servisa pumpne stanice

Izvor: Autor

Ukoliko servis nije odrađen, on se također prikazuje na početnoj stranici prijavljenog korisnika kako bi se omogućio brži pregled.

Klikom na „Pregled ugrađene opreme“ ili „Radovi na objektu“ u izborniku pojedinog objekta pojavljuju se pripadajući popisi zapisa iz baze podataka. Kod dodavanja ovih zapisa također je moguće unijeti bilješku i priložiti datoteke. Zapisi ovih tipova ne pojavljuju se na početnoj stranici.

### 12.2. Troškovi

Ukoliko se odaberu troškovi u modulu vodoopskrbe, prikazuje se dodatni izbornik sa sljedećim stavkama:

- električna energija
- komunikacije.

Klikom na „Električna energija“ ili „Komunikacije“ otvara se pripadajući popis troškova za sve objekte vodoopskrbe te je pretraživač postavljen na trenutni mjesec. Putem pretraživača također se mogu filtrirati objekti za koje se žele prikazati troškovi te odabrati period u kojem su troškovi upisani.

Ispod popisa troškova ispisan je zbroj ukupnih troškova za odabrani period i objekte.

## Troškovi električne energije - Vodoopskrba

PRETRAŽIVANJE

Objekti

Od  Do

☰ Popis troškova el. energije

Datum	Objekt	Broj OMM	Mrežarina(Kn)	Jalova(Kn)	El.energija(Kn)	RVT	RNT	
18.01.2017	PS15 Dunjkovec	123456	182,60	533,53	735,87	56	76	✖
UKUPNO ZAPISA: 1			182,60	533,53	735,87	56	76	

Slika 16. Prikaz troškova električne energije za vodoopskrbu

Izvor: Autor

U ovom dijelu aplikacije, osim standardnog način unosa novih podataka, također postoji i način za masovno dodavanje zapisa klikom na gumb „++“.

Prilikom masovnog dodavanja zapisa troškova komunikacije otvara se popis objekata koji u administratorskom dijelu imaju naznačeno da imaju te troškove te se automatski ispunjavaju brojevi telefona i napomene.

Prilikom masovnog dodavanja zapisa troškova električne energije otvara se popis objekata koji u administratorskom dijelu imaju naznačeno da imaju te troškove te se automatski ispunjava podatak o obračunskom mjernom mjestu.



## Troškovi komunikacija - Vodoopskrba

Dodavanje novih troškova

Mjesec

Objekt	Broj telefona	Napomena	SMS/MB	Cijena
Crpilište Prelog	<input type="text" value="098/654-321"/>	<input type="text" value="Serijski broj kartice 654321"/>	<input type="text"/>	<input type="text"/>
Crpilište Nedelišće	<input type="text" value="098/123-456"/>	<input type="text" value="Serijski broj kartice 123456"/>	<input type="text"/>	<input type="text"/>

Slika 17. *Primjer masovnog dodavanja zapisa troškova komunikacija*

*Izvor: Autor*

### 12.3. Plan održavanja

Klikom na „Plan održavanja“ u izborniku vodoopskrbe otvara se popis planova održavanja s pretraživačem postavljenim na trenutni mjesec.

## Plan održavanja - Vodoopskrba

PRETRAŽIVANJE

Od  Do

☒ Evidencija planova održavanja

**23.01.2017**

BILJEŠKA:

Ugradnja novih pumpi u Zdenac 3.

Slika 18. *Prikaz planova održavanja za vodoopskrbu*

*Izvor: Autor*

Kod unosa novog ili uređivanja postojećeg plana održavanja moguće je unijeti bilješku, kao i učitati popratne datoteke. Planovi održavanja mogu se kao i servisi kod pojedinih objekata postaviti da su odrađeni. Ukoliko plan nije odrađen on se pojavljuje na početnoj stranici radi bržeg pregleda.

### 13. MODUL „ODVODNJA“

Modul „odvodnja“ ima svu funkcionalnost kao i modul „Vodoopskrba“, kao i dodatnu mogućnost koja omogućava unos ispitivanja i pregleda kod pojedinih objekata.

Klikom na „Ispitivanja i pregled“ kod izbornika pojedinog objekta prikazuje se popis obavljenih ispitivanja i pregleda. Kao i kod mjerenja u pretraživaču ovdje postoji popis parametara po kojima se želi filtrirati prikaz podataka. Prikaz podataka i ispis je također dinamičan na način da se na popisu prikazuju samo odabrani parametri.

#### PS14 Dunjkovec

Mjerenja
Servisi
Pregled ugrađene opreme
Radovi na objektu
Ispitivanja i pregled

PRETRAŽIVANJE

Ispitivanja

\* Instalirana snaga ili limitator (A)
\* Komunikacija (GPRS,SMS)
\* Osigurač pumpe
\* Skala ampermetra
\* Verzija programa telemetrije

Od 01.09.2015 Do 10.09.2017 Traži

Popis obavljenih ispitivanja

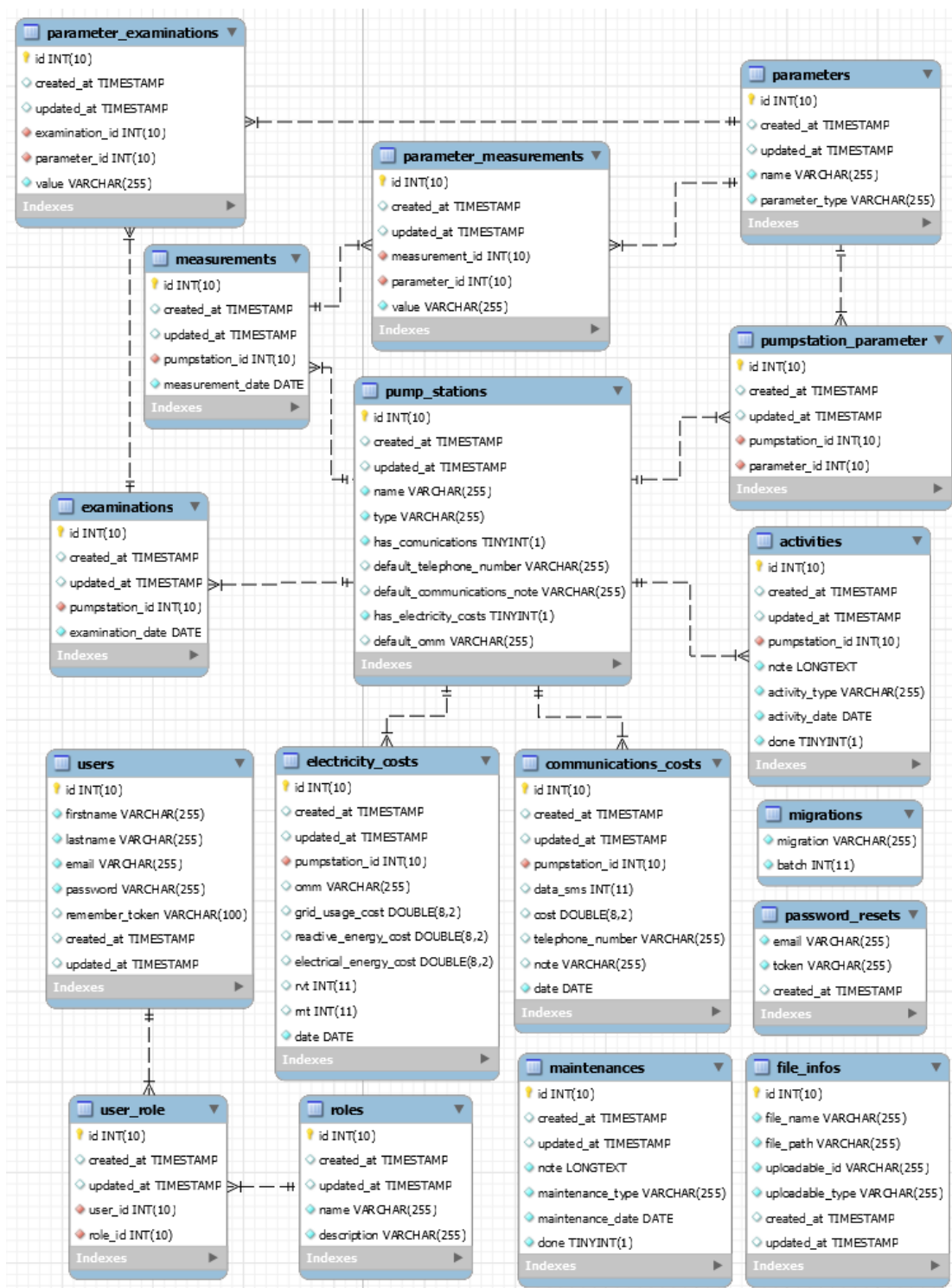
Datum	Instalirana snaga ili limitator (A)	Komunikacija (GPRS,SMS)	Osigurač pumpe	Skala ampermetra	Verzija programa telemetrije	
24.05.2016	25A	SMS Pročistač Čakovec	C16A	0-25A	Pročistač Čakovec v1.0	✗
18.01.2016	25A	-	C16A	0-25A	-	✗

Slika 19. Prikaz obavljenih ispitivanja na pumpnoj stanici

Izvor: Autor

## 14. BAZA PODATAKA

### 14.1. Dijagram baze podataka



Slika 20. Dijagram baze podataka

Izvor: Autor

**14.2. Tablica „users“**

U tablicu „users“ upisuju se osobni podatci kao i kriptirane lozinke svih korisnika aplikacije. Inicijalno sadrži jedan zapis s početnim podacima administratora.

**14.3. Tablica „roles“**

Tablica „roles“ sadrži popis uloga koje mogu biti dodijeljene korisnicima. Ova tablica dolazi unaprijed popunjena te u aplikaciji ne postoji mogućnost mijenjanja ili dodavanja zapisa.

**14.4. Tablica „user\_role“**

Tablica „user\_role“ služi kao međutablica koja povezuje korisnike aplikacije s ulogama. Sadržava strani ključ na tablicu „users“ i tablicu „roles“.

**14.5. Tablica „password\_resets“**

Tablica „password\_resets“ služi za resetiranje korisničkih lozinki, a automatski je kreirana od strane aplikacije. Kako u aplikaciji samo administrator ima pravo promjene korisničkih podataka, ova tablica se ne koristi.

**14.6. Tablica „pump\_stations“**

U tablicu „pump\_stations“ upisuju se osnovni podatci o objektima vodoopskrbe i odvodnje vode koje kreira korisnik.

**14.7. Tablica „parameters“**

U tablicu „parameters“ upisuju se svi parametri koji se kreiraju za potrebe ispitivanja ili mjerenja kod pumpnih stanica. Zapise u ovoj tablici kreira krajnji korisnik putem aplikacije.

**14.8. Tablica „pumpstation\_parameter“**

Tablica „pumpstation\_parameter“ služi kao međutablica koja povezuje pumpne stanice s parametrima za mjerenje ili ispitivanje. Sadržava strani ključ na tablicu „pump\_stations“ i tablicu „parameters“.

**14.9. Tablica „measurements“**

U tablicu „measurements“ upisuju se sva mjerenja parametara koja korisnici obavljaju na pumpnim stanicama. Sadržava strani ključ na tablicu „pumpstation\_examinations“.

**14.10. Tablica „parameter\_measurements“**

Tablica „pumpstation\_measurements“ služi kao međutablica koja povezuje parametre za mjerenja s mjerenjima. Sadržava strani ključ na tablicu „parameters“ i tablicu „measurements“.

**14.11. Tablica „activities“**

U tablicu „activities“ upisuju se sve evidencije servisa, pregleda ugrađene opreme i radova na nekom objektu. Tablica sadržava strani ključ na tablicu „pump\_stations“.

**14.12. Tablica „examinations“**

U tablicu „examinations“ upisuju se sva ispitivanja parametara koja korisnici obavljaju na pumpnim stanicama. Sadržava strani ključ na tablicu „pump\_stations“.

**14.13. Tablica „parameter\_examinations“**

Tablica „pumpstation\_examinations“ služi kao međutablica koja povezuje parametre za ispitivanje s ispitivanjima. Sadržava strani ključ na tablicu „parameters“ i tablicu „examinations“.

**14.14. Tablica „electricity\_costs“**

U tablicu „electricity\_costs“ upisuju se troškovi električne energije pumpnih stanica. Tablica sadržava strani ključ na tablicu „pump\_stations“.

**14.15. Tablica „communications\_costs“**

U tablicu „electricity\_costs“ upisuju se troškovi komunikacija pumpnih stanica po brojevima telefona. Tablica sadržava strani ključ na tablicu „pump\_stations“.

**14.16. Tablica „maintenances“**

U tablicu „maintenances“ upisuju se planovi za održavanje na vodoopskrbnim sustavima ili sustavima odvodnje.

**14.17. Tablica „migrations“**

Tablica „migrations“ služi kao pomoćna tablica za verzioniranje baze podataka. U njoj su zapisani podaci o trenutnoj migraciji kao i svim prijašnjim migracijama.

**14.18. Tablica „file\_infos“**

U tablicu „file\_infos“ zapisuju se podaci o nazivima i putanjama datoteka koje je korisnik učitao u aplikaciju.

## 15. SIGURNOST

U aplikaciju su ugrađene mjere kojima se poboljšava sigurnost i ispravnost rada kao i otpornost na razne zlonamjerne napade.

### 15.1. Autentifikacija

Laravel framework dolazi sa svojim vlastitim osnovnim sustavom autentifikacije koji je naknadno nadograđen s podrškom za prava pristupa korisnika.

Kako korisnici ne bi mogli pristupiti sadržaju za koji nisu ovlašteni, u aplikaciju je uveden međusloj koji sprječava pristup rutama kojima korisnik nema pravo pristupa. Popis ruta te korisničkih uloga koje mogu pristupiti tim rutama nalazi se u datoteci „routes/web.php“.

Kako bi se spriječili „brute-force“ pokušaji otkrivanja lozinki korisnika, u Laravel framework ugrađen je mehanizam koji sprječava prijavu korisnika na 60 sekundi ukoliko se 5 puta unese kriva lozinka.

### 15.2. Hashiranje lozinke

Kako bi se spriječilo otkrivanje lozinki svih korisnika ukoliko bi netko neovlašten dobio pristup bazi podataka, iste su hashirane korištenjem „bcrypt“ funkcije. Funkcija za hashiranje bazirana je na „Blowfish“ simetričnom kriptografskom algoritmu kojeg karakterizira jednostavnost implementiranja i velika brzina izvođenja [14]. Lozinka „hashirana“ „bcrypt“ funkcijom vrlo je otporna na „brute-force“ pokušaje napada kao i na „rainbow table“ napade koji se baziraju na tablici koja služi za preokretanje kriptografskih hash funkcija [15]. Budući da je hashiranje jednostrani proces, hashirana lozinka se više ne može dekriptirati te se kod prijave korisnika uspoređuju hashirane lozinke iz baze podataka i korisničkog zahtjeva.

### 15.3. CSRF Token

CSRF (engl. *Cross-Site Request Forgery*) vrsta je zlonamjernog napada kod kojeg napadač želi uzrokovati nenamjernu akciju u ime ovlaštenog korisnika poput resetiranja lozinke, brisanja ili uređivanja podataka na način da se iskorištava ranjivost stranice koja ne provjera izvor HTTP zahtjeva prije obrade i izvedbe [16].

Za sprječavanje te vrste napada generira se „CSRF token“ koji se ugrađuje u svaku formu te se pomoću njega potvrđuje da je korisnik upravo taj koji šalje zahtjev na poslužitelj.

```
<form role="form" method="POST" action="{{ url('/login') }}">
    {{ csrf_field() }}
</form>
```

Kod 14. *Primjer koda za kreiranje CSRF tokena ugrađenog u HTML formu*

*Izvor: Autor*

## 15.4. Kontrola unosa podataka

U aplikaciju je ugrađen mehanizam kontrole unosa podataka kako bi svi podaci uneseni u bazu podataka uvijek bili ispravni i u ispravnom formatu te kako bi obavezni podatci uvijek bili uneseni.

### Dodavanje nove pumpne stanice

Naziv

Polje je obavezno.

Kategorija objekta

Mjerenja

Komunikacijski troškovi

☐

Troškovi el. energije

☐

Povratak

Spremi

Slika 21. *Prikaz poruke za obavezno polje prilikom dodavanja pumpne stanice*

*Izvor: Autor*



## 16. ZAKLJUČAK

Korištenjem Laravel PHP frameworka izrađena je web-aplikacija koja služi za praćenje rada pumpnih stanica u vodoopskrbi i odvodnji. Prilikom razvoja aplikacije korištene su i popratne moderne web-tehnologije koje omogućavaju responzivni dizajn i intuitivno moderno korisničko sučelje.

Najvažnije funkcionalnosti aplikacije obuhvaćaju upravljanje korisnicima, administraciju pumpnih stanica, bilježenje rezultata mjerenja i ispitivanja obavljenih na stanicama, planiranje servisa, pregleda ugrađene opreme kao i radova na objektima te izradu planova za održavanje na cjelokupnom sustavu vodoopskrbe ili odvodnje, kao i bilježenje troškova električne energije i komunikacija.

Sve podatke u aplikaciji moguće je prikazati u PDF obliku te je na taj način omogućeno ispisivanje putem pisača.

U aplikaciji postoji više tipova korisnika što omogućava radnicima dodavanje i uređivanje podataka vezanih uz pumpne stanice, ali omogućava i kreiranje korisnika koji mogu samo pregledavati podatke ukoliko je potreban uvid u rad od strane nadgledatelja.

Kako bi se osigurao siguran i ispravan rad aplikacije, ugrađene su mjere koje štite korisničke podatke, mjere koje štite korisnika od zlonamjernih napada te mehanizmi za provjeru valjanosti podataka.

Aplikacija je za korištenje namijenjena tvrtkama koje se bave isporukom usluga vodoopskrbe te odvodnje onečišćene vode.

## 17. POPIS LITERATURE

- [1] Crpne stanice,  
[https://www.grad.unizg.hr/\\_download/repository/1.6.\\_Crpne\\_stanice.pdf](https://www.grad.unizg.hr/_download/repository/1.6._Crpne_stanice.pdf) (25.08.2016.)
- [2] 6S-252/6S-253–SEMI-AXIAL ELECTRIC PUMPS,  
<http://www.saerelettropompe.com/product/6s-252-6s-253-elettropompe-semiassiali/> (25.08.2016.)
- [3] Basic MVC Architecture,  
[https://www.tutorialspoint.com/struts\\_2/basic\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm) (25.08.2016.)
- [4] HTML, <https://www.w3schools.com/html/> (25.08.2016.)
- [5] Uvod u PHP, <http://php.com.hr/77> (25.08.2016.)
- [6] PHP The Right Way, <http://www.phptherightway.com/> (25.08.2016.)
- [7] CSS Introduction, [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp) (25.08.2016.)
- [8] JavaScript Introduction, [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp) (25.08.2016.)
- [9] MySQL documentation, <https://dev.mysql.com/doc/> (25.08.2016.)
- [10] Code editing. Redefined., <https://code.visualstudio.com/> (25.08.2016.)
- [11] Laravel 5.3 documentation, <https://laravel.com/docs/5.3> (25.08.2016.)
- [12] Laracasts, <https://laracasts.com/> (27.08.2016.)
- [13] Bootstrap, <https://getbootstrap.com> (26.08.2016.)
- [14] bcrypt, <http://bcrypt.sourceforge.net/> (07.04.2017.)
- [15] Rainbow Tables: Your Password's Worst Nightmare  
<https://lifewire.com/rainbow-tables-your-passwords-worst-nightmare-2487288>  
(07.02.2017.)
- [16] CSRF napadi, <http://www.cert.hr/node/15454> (07.04.2017.)

**POPIS SLIKA**

Slika 1.	Visokotlačna bunarska potopna crpka.....	10
Slika 2.	Shema položaja crpne stanice na kanalu .....	12
Slika 3.	Uspješno kreiranje osnovne autentifikacije u Laravel frameworku.....	25
Slika 4.	Početna stranica aplikacije prijavljenog korisnika .....	30
Slika 6.	Izbornik modula „Administracija“ .....	31
Slika 7.	Prikaz popisa parametara u aplikaciji.....	32
Slika 8.	Dodavanje novog parametra.....	32
Slika 9.	Prikaz popisa objekata u modulu „administracija“ .....	33
Slika 10.	Uređivanje postojeće pumpne stanice .....	34
Slika 11.	Pregled popisa korisnika aplikacije .....	35
Slika 12.	Dodavanje novog korisnika u aplikaciju .....	35
Slika 13.	Prikaz popisa obavljenih mjerenja na pumpnoj stanici .....	37
Slika 14.	Prikaz evidencije servisa pumpne stanice .....	37
Slika 15.	Uređivanje postojećeg zapisa evidencije servisa pumpne stanice .....	38
Slika 16.	Prikaz troškova električne energije za vodoopskrbu .....	39
Slika 17.	Primjer masovnog dodavanja zapisa troškova komunikacija .....	40
Slika 18.	Prikaz planova održavanja za vodoopskrbu .....	40
Slika 19.	Prikaz obavljenih ispitivanja na pumpnoj stanici.....	41
Slika 20.	Dijagram baze podataka .....	42
Slika 21.	Prikaz poruke za obavezno polje prilikom dodavanja pumpne stanice .....	47

**POPIS KODOVA**

Kod 1.	Primjer ispisa teksta u PHP programskom jeziku .....	14
Kod 2.	Primjer referenciranja vanjske CSS datoteke u HTML jeziku.....	15
Kod 3.	Primjer CSS pravila za bojanje teksta crvenom bojom i centriranje.....	15
Kod 4.	Primjer referenciranja vanjske JavaScript datoteke u HTML jeziku .....	16
Kod 5.	Primjer naredbe za odabir svih korisnika u MySQL jeziku .....	17
Kod 6.	Primjer naredbe za dodavanje Bootstrap front-end frameworka u projekt pomoću komandne linije .....	18
Kod 7.	Primjer dodavanja Bootstrap front-end frameworka u projekt ručnim uređivanjem „composer.json“ datoteke .....	19
Kod 8.	Primjer naredbe za preuzimanje Laravel instalacije .....	21
Kod 9.	Migracija za kreiranje tablice za mjerenja parametara pumpnih stanica .....	26
Kod 10.	Model klasa „Activity“ koja predstavlja servise, radove ili preglede ugrađene opreme na pumpnim stanicama .....	27
Kod 11.	HTML kod „master“ pogleda.....	28
Kod 12.	Primjer kontrolera za prikaz podataka određenog korisnika.....	29
Kod 13.	Rute za upravljanje korisnicima.....	29
Kod 14.	Primjer koda za kreiranje CSRF tokena ugrađenog u HTML formu .....	47